On Average Throughput and Alphabet Size in Network Coding

Chandra Chekuri, Christina Fragouli, and Emina Soljanin

Abstract

We examine the throughput benefits that network coding offers with respect to the average throughput achievable by routing, where the average throughput refers to the average of the rates that the individual receivers experience. We relate these benefits to the integrality gap of a standard LP formulation for the directed Steiner tree problem. We describe families of configurations over which network coding at most doubles the average throughput, and analyze a class of directed graph configurations with N receivers where network coding offers benefits proportional to $\sqrt{N}$. We also discuss other throughput measures in networks, and show how in certain classes of networks, the average throughput can be achieved uniformly by all receivers by employing vector routing and channel coding. Finally, we show configurations where use of randomized coding may require an alphabet size exponentially larger than the minimum alphabet size required.

Index Terms

Network coding, multicast, routing, throughput.

This work was supported in part by DIMACS, NSF under award No. CCR-0325673, and FNS under award No. 200021-103836/1. The material in this paper was presented in part at Allerton Conference on Communication, Control, and Computing, Urbana, IL, USA, Oct. 2004.

C. Chekuri and E. Soljanin are with Bell Labs, Email: {chekuri,emina}@lucent.com. C. Fragouli is with EPFL, Email: christina.fragoulli@epfl.ch.

I. INTRODUCTION

Consider a communication network represented as a directed graph $G = (V, E)$ with unit capacity edges, and h unit rate information sources $S_1, \ldots, S_h$ that simultaneously transmit information to N receivers $R_1, \ldots, R_N$ located at distinct nodes. Assume that the min-cut between the sources and each receiver node is h. The max-flow, min-cut theorem states that, if receiver R_i could utilize the network resources by itself, it would be able to receive information at rate h.

Recently it has been realized that allowing nodes in communication networks to re-encode the information they receive in addition to re-routing, increases the capacity of the network. This type of coding is termed network coding [1], [2]. In fact it was shown that by linear re-encoding, the min-cut rate can be achieved in multicasting to several sinks [1], [2]. That is, network coding allows each receiver to retrieve information at rate h, even when N receivers share the network resources. This is generally not the case when we use routing, *i.e.,* when we allow intermediate nodes only to forward and not to code. Thus network coding can offer throughput benefits as compared to routing.

A natural question to ask is how large these throughput benefits are. Let $T_c = h$ denote the rate that the receivers experience when network coding is used. We consider the following types of routing: *integral routing*, which implies that through each unit capacity edge we can route one unit rate source, and *fractional routing*, which implies that through each edge we can route fractional rates of different sources. Under a given integral routing scheme, let T_i^j denote the rate that receiver j experiences. Similarly let T_f^j be the rate that receiver j experiences under a given fractional routing scheme. We let $T_i = \max\min_{j=1\ldots N}\{T_i^j\}$ and $T_f = \max\min_{j=1\ldots N}\{T_f^j\}$ denote the maximum integral and fractional rate we can route to all receivers, where N is the number of receivers. The minimization is over the rates the individual receivers experience, and the maximization over all possible routing strategies. The benefits that network coding can offer as compared to routing are quantified by the ratios $\frac{T_i}{T_c}$ and $\frac{T_f}{T_c}$.

In [3] it was shown that, for undirected graphs, if we allow fractional routing, the throughput benefit that network coding offers over routing is bounded by a factor of two, *i.e.,* $\frac{T_f}{T_c} \leq 2$. Experimental results in [4] over the network graphs of six Internet service providers also showed the small throughput benefits in this case.

This result does not transfer to directed graphs. The authors in [5] provide an example of a directed graph where the integral throughput benefits scale proportionally to the number of sources, namely, $\frac{T_i}{T_c} = \frac{1}{h}$. We show in this paper that a similar result is true even if we allow fractional routing. In other words, if we compare the common rate guaranteed to all receivers under routing with the rate that network coding can offer, the benefits network coding offers are proportional to the number of sources h.

In [6] it was shown that, for both directed and undirected graphs, $\frac{T_c}{T_f}$ equals the integrality gap of a standard linear programming formulation for the directed Steiner tree problem. Known lower bounds on the integrality gap for directed graphs are $\Omega(\sqrt{N})$ [7] and $\Omega((\log n / \log\log n)^2)$ [8] where n is the number of nodes in the underlying graph. For undirected graphs, a known gap is $\frac{8}{7}$ (see [6] for the example).

In this paper we focus on the throughput benefits network coding offers when multicasting to a set of receivers that have the same min-cut. Work in the literature has also started examining throughput benefits that network coding can offer for other types of traffic, see for example [3], [9], and [10].

Even for the case of multicasting, there is still a very limited understanding on what are the structural properties of multicast configurations that necessitate the use of network coding to achieve the min-cut rate to each receiver. In order to increase our understanding in this aspect, we relax the requirement that routing has to convey the same rate to all the receivers of the multicast session, and compare the sum rate we can achieve with and without network coding. That is, we examine the average throughput achieved with integral and fractional routing, $T_i^{av} = \max \frac{1}{N} \sum_{j=1 \ldots N} T_i^j$ and $T_f^{av} = \max \frac{1}{N} \sum_{j=1 \ldots N} T_f^j$ respectively, where the averaging is performed over the rate that each individual receiver experiences. The set of configurations where the average rate achieves a constant factor of the min-cut is much larger than the set of configurations where the common rate guaranteed to all receivers can be made a constant factor of the min-cut. For example, as we will discuss in Section IV, for the multicast configuration in [5], $T_c = h$, $T_i = 1$ while $T_i^{av} \geq \frac{h}{2}$ where h is the number of sources. By decoupling the problem of achieving a high sum rate, from the problem of balancing the rate towards the different receivers, we hope to increase our intuition of when network coding offers throughput benefits from a theoretical point of view.

Moreover, from a practical point of view, for applications that are robust to loss of packets such as real time audio and video, the average throughput achieved with routing might be a more appropriate measure of performance to compare against network coding. This is especially true when the number of receivers is large and the throughput they experience tends to concentrate around the average value. This is the case in the example in [5]. In fact, multicast sessions where the different receivers experience different rates is the majority of cases in practical scenarios, and coding schemes (ex, Fountain codes [11], [12]) have been developed to address this situation. For the example in [5] we will describe such a coding scheme that exploits the $\frac{h}{2}$ average rate to convey $\frac{h}{2}$ common information rate to all receivers. We will then present a method which combines vector routing and erasure correcting codes to translate the average to common throughput for an arbitrary multicast configuration. This method can be thought of as a generalization of the vector routing capacity in [13].

The contributions of this paper also include the following. We describe a Linear Programming (LP) formulation for calculating T_f^{av} over directed graphs that performs packing of partial Steiner trees. Using this formulation we show that the average throughput benefits of network coding can be related to the integrality gap of a standard LP formulation for the directed Steiner tree problem.

For N much larger than h, the behavior of T_f^{av} and T_f can be quite different. We describe a number of configurations, that include the example in [5], where although network coding can offer significant benefits as compared to T_f, i.e., $\frac{T_f}{T_c}$ can become arbitrarily small, it can only offer a constant factor benefit with respect to the average rate T_f^{av}.

We then describe and analyze a class of directed graph configurations where network coding offers significant benefits as compared to the average throughput. These configurations were originally constructed in [7] to obtain a lower bound on the integrality gap for the directed Steiner tree problem. We show that employing network coding over this class of directed graphs can offer throughput benefits proportional to $\sqrt{N}$, where N is the number of receivers, with regard to the average (and as a result to the common) throughput, i.e., $\frac{T_f}{T_c} \leq \frac{T_f^{av}}{T_c} \leq \frac{1}{\sqrt{N}}$.

These graphs also illustrate that use of randomized coding may require an alphabet size exponentially larger than the minimum alphabet size required. The idea in randomized network coding [5], [14], [15] is to randomly combine over a finite field the incoming information flows and show that the probability of error can become arbitrarily small as the size of the finite field increases. We show that for this class of configurations, to guarantee a small probability of error, we may need to use an exponentially large alphabet size. In contrast, we prove that a binary alphabet size is in fact sufficient for network coding. We construct a deterministic algorithm that has linear complexity, can be used to perform network coding over this class of configurations, and requires binary alphabet. This coding scheme effectively transforms the configuration in [7] to a *bipartite* configuration, *i.e.,* a configuration where network coding is performed only on information streams carrying the source symbols.

The paper is organized as follows. In Section II we briefly introduce our notation. Section III presents linear programming formulations and results. Section V discusses a number of configurations where network coding offers benefits as compared to the common throughput. Section IV discusses a family of configurations where network coding also offers benefits as compared to the common throughput. Section VI concludes the paper.

II. Network Models and Problem Formulation

We consider a communications network represented by a directed acyclic graph $G = (V, E)$ with unit capacity edges. There are h unit rate information sources $S_1, \ldots, S_h$ and N receivers $R_1, \ldots, R_N$. There are h edge disjoint paths from the h sources to each receiver. For receiver j, we denote these paths as (S_i, R_j), $i = 1, \ldots, h$. The h information sources multicast information simultaneously to all N receivers at rate h.

We are interested in the throughput benefits that network coding can offer as compared to routing (uncoded transmission). Let T_c denote the rate that the receivers experience when network coding is used. We will use the following notation for the routing throughput.

- T_i^j and T_f^j denote the rate that receiver j experiences with fractional and integral routing respectively under a specific routing strategy.
- $T_i = \max \min_{j=1\ldots N}\{T_i^j\}$ and $T_f = \max \min_{j=1\ldots N}\{T_f^j\}$ denote the maximum integral and fractional rate we can route to all receivers, where the maximization over all possible routing strategies.
- $T_i^{av} = \frac{1}{N} \max \sum_{j=1}^{N} T_i^j$ and $T_f^{av} = \frac{1}{N} \max \sum_{j=1}^{N} T_f^j$ denote the maximum integral and fractional *average* throughput. We will use T^{av} to discuss results that apply both to integral and fractional average routing.

The benefits that network coding can offer over a configuration with respect to the common throughput can be quantified by the quantities

$$\frac{T_i}{T_c} \text{ and } \frac{T_f}{T_c}.$$

The problem of calculating T_f (T_i) is equivalent to the problem of packing fractional (integral) trees that are rooted at the source nodes and span the set of receivers.

In this paper we are mainly interested in comparing the average throughput when network coding is used to the average throughput when only routing transmission is allowed. Equivalently, we will be comparing the sum rate achieved with and without network coding. The throughput benefits that network coding offers as compared to the average throughput can be quantified as

$$\frac{T_i^{av}}{T_c} \text{and } \frac{T_f^{av}}{T_c}.$$

The problem of calculating T_f^{av} (T_i^{av}) is equivalent to the problem of packing fractional (integral) *partial* Steiner trees, *i.e.,* trees that are rooted at the source nodes that span a subset of the receivers.

For a multicast configuration with h sources and N receivers, it holds that

$$T_c = h,$$

from the main network multicast theorem [1], [2]. Also, because there exists a tree spanning the source and the receiver nodes, the uncoded throughput is at least N. We, therefore, have

$$1 \leq T_i^{av} \leq T_f^{av} \leq h,$$

and thus

$$\frac{1}{h} \leq \frac{T_i^{av}}{T_c} \leq \frac{T_f^{av}}{T_c} \leq 1. \tag{1}$$

The upper bound in (1) is achievable by the configurations in which network coding is not necessary for multicast. Much less in known about the lower bound on the ratio T_i^{av}/T_c. We here find lower bounds to this quantity for several classes of networks, where classification of networks is performed based on their information flow decomposition described in [16].

The information flow decomposition partitions the network into subgraphs through which the same information flows, i.e., "processing" happens only on subgraph boundaries. Each such part is a tree, that is rooted either at the source, or at nodes where we might need to perform coding operations. For the network code design problem, the structure of the network inside these trees does not play any role; we only need to know how the trees are connected and which receiver nodes observe the information that flows in each tree. Thus, we can contract each tree to a single vertex, and get a graph whose nodes correspond to entire areas of the original network. We call this process and the resulting graph the information flow decomposition of the network.

In the information flow decomposition graph, there are nodes with no incoming edges, called sources (or source nodes), and nodes with two or more in-going edges called coding nodes. We say a node *contains* R_j to indicate that receiver R_j *observes* that node (flow), and label the node accordingly. Note that each receiver observes h nodes in the information flow graph. An example of a network and its information flow decompositions is given in Fig. $1(a)-(b)$. There exist two source nodes and five coding nodes; each of the 10 receivers observes two coding nodes.

We are in particular interested in information flow graphs that are *minimal* with the min-cut property, namely those for which removing any edge would violate the min-cut property for at least one receiver. A minimal information flow graph for the network in Fig. $1(a)$ is depicted in Fig. $1(c)$. The procedure for information flow decomposition for a network is described in

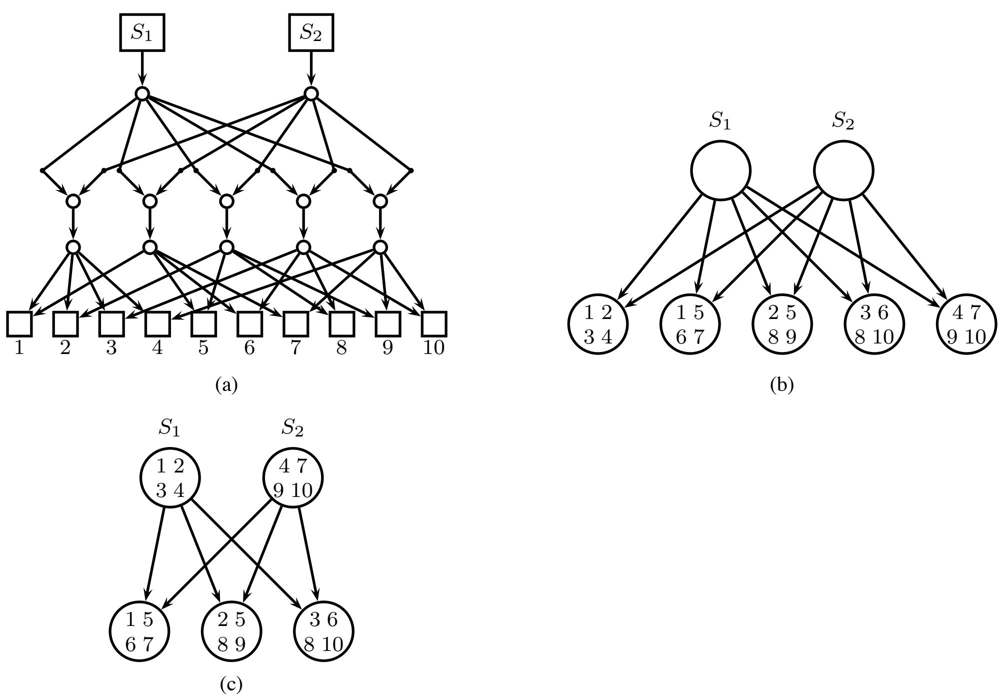


Fig. 1. (a) A network with two sources and 10 receivers; (b) an information flow decomposition of the network, and (c) a minimal information flow graph.

detail in [16].

Note that in Fig. 1 each coding point has only source nodes as its parents, *i.e.*, network coding is performed only on information streams carrying the source symbols. We refer to this type of information flow graph as a *bipartite configuration*.

III. LP FORMULATIONS

In this section we consider a directed graph $G = (V, E)$, a root (source) vertex $S \in V$, and a set $\mathcal{R} = \{R_1, R_2, \ldots, R_N\}$ of N terminals (receivers) which we describe together as an instance $\{G, S, \mathcal{R}\}$. With every edge e of the graph we can in general associate two parameters, a capacity $c_e \geq 0$, and a cost (weight) $w_e \geq 0$. Let $c = [c_e]$ and $w = [w_e]$, $e \in E$ denote vectors that collect the set of edge capacities and edge weights respectively. Depending on the problem, the edge weights or the edge capacities or both are relevant.

In the *Steiner tree* problem, we are given an instance $\{G, S, \mathcal{R}\}$ and a set of non-negative edge weights w. We are asked to find the minimum weight tree that connects the source to all the terminals. Here edge capacities are not relevant: the Steiner tree either uses or does not use an edge.

We call a set of vertices $\mathcal{D} \subset V$ *separating,* if $\mathcal{D}$ contains the source vertex S and $V \setminus \mathcal{D}$ contains at least one of the terminals in $\mathcal{R}$. Let $\delta(\mathcal{D})$ denote the set of edges from $\mathcal{D}$ to $V \setminus \mathcal{D}$, that is, $\delta(\mathcal{D}) = \{(u, v) \in E : u \in \mathcal{D}, v \notin \mathcal{D}\}$. We consider the following formulation for the Steiner tree problem

$$\min \sum_{e \in E} w_e x_e$$

$$\sum_{e \in \delta(\mathcal{D})} x_e \geq 1, \quad \forall\ \mathcal{D}\text{: } \mathcal{D} \text{ is separating}$$

$$x_e \in \{0, 1\}, \quad \forall\ e \in E$$

where there is a variable x_e for each edge $e \in E$ to indicate whether the edge is used in the tree or not. Note that any vector $x = \{x_e, e \in E\}$ satisfying the constraints of the above LP can be interpreted as a set of capacities for the edges of G, and that the constraints then ensure that the min-cut from the source S to each receiver in the capacitated graph (G, S, x) is at least one. Let $\text{OPT}(G, w, S, \mathcal{R})$ be the value optimum solution for the given instance.

In the above formulation, the objective function and the constraints are linear in the underlying variables. Further, the variables are constrained to be integers. Such a formulation is referred to as an integer program (IP). If all the variables can take on values from the domain of real numbers we obtain a linear program (LP). Please see [17, Parts 3 and 4] for more details on integer and linear programs. It is easy to see that the constraints in the above integer program are necessary for the Steiner tree problem. It is less obvious that they are sufficient but this can be shown by some elementary graph theoretic arguments. We give a brief sketch below. Consider a feasible solution to the integer program and let $E' \subset E$ be the set of edges e such that $x_e = 1$. Let $G' = (V, E')$ the graph induced by E'. Consider the set D of all receivers that can be reached from S in G'. If D does not include all the receivers then it can be seen that D is a separating set with no edge crossing it and hence contradicts the feasibility of the solution x. This ensure that in G' there is a path from S to every receiver. A minimal subset of E' that ensures connectivity from S to every receiver can be shown to be a tree. Thus we conclude that any optimum feasible solution of the above integer program does indeed induce a Steiner tree. The formulation above has an exponential number of constraints; however, there is an equivalent compact formulation with a polynomial number of constraints and variables. This equivalence relies on the well-known maxflow-mincut theorem for single-commodity flows. We refer the reader to [18, Ch. 9] for more details.

The linear relaxation of the above IP is obtained by replacing the constraints $x_e \in \{0, 1\}$, $e \in E$ by $0 \le x_e \le 1$, $e \in E$. We can further simplify this to $x_e \ge 0$, $e \in E$, by noticing that if a solution is feasible with $x_e \ge 1$, then it remains feasible by setting $x_e = 1$. For a given instance $(G, S, \mathcal{R})$, let $\textsc{lp}(G, w, S, \mathcal{R})$ denote the optimum value of the resulting linear program on the instance. The value $\textsc{lp}(G, w, S, \mathcal{R})$ lower bounds the cost of the integer program solution $\textsc{opt}(G, w, S, \mathcal{R})$. The *integrality gap* of the formulation on G is defined as

$$\alpha(G, S, \mathcal{R}) = \max_{w} \frac{\textsc{opt}(G, w, S, \mathcal{R})}{\textsc{lp}(G, w, S, \mathcal{R})},$$

where the maximization is over all possible edge weights. Note that $\alpha(G)$ is invariant to scaling weights.

Let w^* be the set of edge weights that achieves the maximum value $\alpha(G, S, \mathcal{R})$, and $x^* = \{x^*_e, e \in E\}$ be an optimum solution for the associated LP. In [6] it was shown that, if we consider the instance $\{G, S, \mathcal{R}\}$, associate capacity $c_e = x^*_e$ with each edge e, and compare the throughput we can get with and without network coding (T_c and T_f respectively) on this capacitated graph, then $\alpha(G,S,\mathcal{R}) = \frac{\text{OPT}(G,w^*,S,\mathcal{R})}{\text{LP}(G,w^*,S,\mathcal{R})} = \frac{T_c(G,c=x^*,S,\mathcal{R})}{T_f(G,c=x^*,S,\mathcal{R})}$. Note that this does not imply that $\text{OPT}(G,w^*,S,\mathcal{R}) = T_c(G,c=x^*,S,\mathcal{R})$ and $\text{LP}(G,w^*,S,\mathcal{R}) = T_f(G,c=x^*,S,\mathcal{R})$. In general, it was shown in [6] that given an instance $\{G,S,\mathcal{R}\}$, $\max_w \frac{\text{OPT}(G,w,S,\mathcal{R})}{\text{LP}(G,w,S,\mathcal{R})} = \max_c \frac{T_c(G,S,\mathcal{R},c)}{T_f(G,S,\mathcal{R},c)}$. That is, for a given multicast configuration $\{G, S, \mathcal{R}\}$, the maximum throughput benefits we may hope to get with network coding will equal the largest integrality gap of the Steiner tree problem possible on the same graph. This result refers to fractional routing; if we restrict our problem to integral routing on the graph, we might get larger throughput benefits.

We now consider the coding advantage for average throughput over a multicast configuration $\{G,S,\mathcal{R}\}$ and a set of non-negative capacities c on the edges of G. We will assume for technical reasons that the min-cut from S to each of the terminals is the same. This can be easily arranged by adding dummy terminals. That is, if the min-cut to a receiver R_i is larger than required, we connect the receiver node to a new dummy terminal through an edge of capacity equal to the min-cut. Then the network coding throughput is given by

$$T_c(G,c,S,\mathcal{R}) = \text{mincut}(S,R_i).$$

The maximum achievable average throughput with routing is given by the maximum fractional packing of *partial* Steiner trees. A partial Steiner tree t stems from the source S and spans all or only a subset of the terminals. With each tree t, we associate a variable y_t denoting a fractional flow through the tree. Let τ be the set of all partial Steiner trees in $\{G, S, \mathcal{R}\}$, and n_t the number of terminals in t. Then the maximum fractional packing of partial Steiner trees is given by the following linear program.

$$\begin{aligned} &\max \sum_{t\in\tau} \frac{n_t}{N} y_t \\ &\sum_{t\in\tau:e\in t} y_t \le c_e, \quad \forall\ e \in E \\ &y_t \ge 0, \quad \forall\ t \in \tau. \end{aligned}$$

Let $T_f^{av}(G,S,\mathcal{R},c)$ denote the value of the above linear program on a given instance. The coding advantage for average throughput on G is given by the ratio

$$\beta(G,S,\mathcal{R}) = \max_c \frac{T_c(G,c,S,\mathcal{R})}{T_f^{av}(G,c,S,\mathcal{R})}.$$

Note that $\beta(G)$ is invariant to scaling capacities. It is easy to see that $\beta(G, S, \mathcal{R}) \geq 1$, since we assumed that the min-cut to each receiver is the same, and thus network coding achieves the maximum possible sum rate. It is also straightforward that $\beta(G, S, \mathcal{R}) \leq \alpha(G, S, \mathcal{R})$, since for any given configuration $\{G, c, S, \mathcal{R}\}$, the average throughput is at least as large as the common throughput we can guarantee to all receivers, i.e., $T_f^{av} \geq T_f$.

Let $\beta(G, S, \mathcal{R}^*)$ denote the maximum average throughput benefits we can get on graph G when multicasting from source S to *any possible subset* of the receivers $\mathcal{R}' \subseteq \mathcal{R}$, i.e.,

$$\beta(G, S, \mathcal{R}^*) = \max_{\mathcal{R}' \subseteq \mathcal{R}} \beta(G, S, \mathcal{R}'). \tag{2}$$

Theorem 1: For a configuration $\{G, S, \mathcal{R}\}$ where $|\mathcal{R}| = N$ receivers and the min-cut to each receiver is the same, we have

$$\beta(G, S, \mathcal{R}^*) \geq \max\{1, \frac{1}{H_N}\alpha(G, S, \mathcal{R})\},$$

where H_N is the Nth harmonic number, namely, $H_N = \sum_{j=1}^{N} 1/j$.

Proof: Consider an instance of a Steiner tree problem $\{G, S, \mathcal{R}\}$ with $|\mathcal{R}| = N$. Let w^* be a weight vector such that

$$\alpha(G, S, \mathcal{R}) = \frac{\text{OPT}(G, w^*, S, \mathcal{R})}{\text{LP}(G, w^*, S, \mathcal{R})} = \max_w \frac{\text{OPT}(G, w, S, \mathcal{R})}{\text{LP}(G, w, S, \mathcal{R})}.$$

Let x^* be an optimum solution for the LP on the instance $(G, w^*, S, \mathcal{R})$. Hence $\text{LP}(G, w^*, S, \mathcal{R}) = \sum_e w_e^* x_e^*$. As we discussed before, we can think of the optimum solution x^* as associating a capacity $c_e = x_e^*$ with each edge e so that the min-cut to each receiver is greater or equal to one, and the cost $\sum_e w_e^* x_e^*$ is minimized.

We are going to examine the average coding throughput benefits we can get on the instance $\{G, c = x^*, S, \mathcal{R}\}$. Since the min-cut to each receiver is at least one, we can achieve throughput $T_c(G, c = x^*, S, \mathcal{R}) \geq 1$. Now, let $y^* = \{y_t^*, t \in \tau\}$ be the optimal fractional packing of partial Steiner trees on $\{G, c = x^*, S, \mathcal{R}\}$. From the definition of $\beta(G, S, \mathcal{R})$, it follows for the capacity vector $c = x^*$, that

$$\beta(G, S, \mathcal{R}) = \max_c \frac{T_c(G, c, S, \mathcal{R})}{T_f^{av}(G, c, S, \mathcal{R})} \geq \frac{T_c(G, c = x^*, S, \mathcal{R})}{T_f^{av}(G, c = x^*, S, \mathcal{R})} \geq \frac{1}{T_f^{av}(G, c = x^*, S, \mathcal{R})} = \frac{1}{\sum \frac{n_t}{N} y_t^*} \tag{3}$$

To further bound $\beta(G, S, \mathcal{R})$, we will find a bound on $\sum \frac{n_t}{N} y_t^*$.

Let $w_t = \sum_{e \in t} w_e^*$ denote the weight of partial tree t, and consider $\sum_{t \in \tau} w_t y_t^*$ (the total weight of the packing y^*). We have

$$\begin{aligned}\sum_{t \in \tau} w_t y_t^* &= \sum_{t \in \tau} w_t \frac{N}{n_t} \cdot y_t^* \frac{n_t}{N} \\ &\geq \min_{t \in \tau} \left\{ w_t \frac{N}{n_t} \right\} \sum_{t \in \tau} y_t^* \frac{n_t}{N}.\end{aligned}$$

Thus there exists a partial tree t_1 of weight w_{t_1} such that

$$w_{t_1} \leq \frac{1}{\sum_{t \in \tau} \frac{n_t}{N} y_t^*} \cdot \frac{n_{t_1}}{N} \sum_{t \in \tau} w_t y_t^*. \tag{4}$$

Moreover, we claim that $\sum_{t \in \tau} w_t y_t^* \leq \sum_{e \in E} w_e^* x_e^*$. Indeed, by changing the order of summation, we get

$$\sum_{t \in \tau} w_t y_t^* = \sum_{t \in \tau} y_t \sum_{e \in t} w_e^* \leq \sum_{e \in E} w_e^* \sum_{t : e \in t} y_t^*.$$

By the feasibility of y^* for the capacity vector x^*, the quantity $\sum_{t : e \in t} y_t^*$ is at most x_e^*. Hence we have that

$$\sum_{t \in \tau} w_t y_t^* \leq \sum_{e \in E} w_e^* x_e^*. \tag{5}$$

From Eq. (3), (4) and (5), it follows that there exists a partial tree t_1 of weight w_{t_1} such that

$$w_{t_1} \leq \beta(G, S, \mathcal{R}) \cdot \frac{n_{t_1}}{N} \sum_{e \in E} w_e^* x_e^*. \tag{6}$$

Now, if $n_{t_1} = N$, then t_1 is a Steiner tree spanning all receivers. From Eq. (6) and definitions of $\beta(G, S, \mathcal{R}^*)$ and $\alpha(G, S, \mathcal{R})$, we get that

$$\beta(G, S, \mathcal{R}^*) \geq \beta(G, S, \mathcal{R}) \geq \frac{w_{t_1}}{\sum_{e \in E} w_e^* x_e^*} \geq \alpha(G, S, \mathcal{R}), \tag{7}$$

which proves the the theorem.

Otherwise, let $\mathcal{R}_{t_1}$ be the $n_1 \neq N$ terminals in t_1, and consider a new instance of the Steiner tree problem obtained by removing terminals $\mathcal{R}_{t_1}$ from $\mathcal{R}$. Note that the solution x^* remains feasible for this new problem. Let $N_2 = |\mathcal{R} \setminus \mathcal{R}_{t_1}| = N - n_1$. We can now repeat the above argument for the instance $\{G, w^*, c^*, S, \mathcal{R} \setminus \mathcal{R}_{t_1}\}$, and, in the same manner, find a new tree t_2 for which a counterpart of (6) holds:

$$w_{t_2} \leq \beta(G, S, \mathcal{R} \setminus \mathcal{R}_{t_1}) \frac{n_{t_2}}{N_2} \sum_{e \in E} w_e^* x_e^* \leq \beta(G, S, \mathcal{R}^*) \frac{n_{t_2}}{N_2} \sum_{e \in E} w_e^* x_e^*.$$

We continue the above process until we cover all terminals by trees, say, $t_1, t_2, \ldots, t_\ell$. Let N_i be the number terminals from $\mathcal{R}$ that remain to be covered before the ith tree is computed. From the above argument, we have that

$$w_{t_i} \le \beta(G, S, \mathcal{R}^*) \frac{n_{t_i}}{N_i} \sum_e w_e^* x_e^*,$$

an thus

$$\sum_{i=1}^{\ell} w_{t_i} \le \beta(G, S, \mathcal{R}^*) \cdot \sum_e w_e^* x_e^* \cdot \sum_{i=1}^{\ell} \frac{n_{t_i}}{N_i}.$$

It is easy to see that

$$\sum_{i=1}^{\ell} \frac{n_{t_i}}{N_i} \le \sum_{i=1}^{N} \frac{1}{N-i+1} = H_N.$$

By construction, the union of the trees $t_1, t_2, \ldots, t_\ell$ contains all the terminals, and thus there is a Steiner tree of weight at most $\sum_i w_{t_i}$. Consequently,

$$\alpha(G, S, \mathcal{R}) = \frac{\text{OPT}(G, w^*, S, \mathcal{R})}{\sum_{e \in E} w_e^* x_e^*} \le \frac{\sum_i w_{t_i}}{\sum_{e \in E} w_e^* x_e^*} \le \beta(G, S, \mathcal{R}^*) H_N.$$

■

Theorem 1 enables us to prove bounds on $\beta(G, S, \mathcal{R}^*)$ using bounds on $\alpha(G, S, \mathcal{R})$. We can think of this theorem as follows. Given $\{G, S, \mathcal{R}\}$, without loss of generality, we can normalize all possible capacities vectors so that $T_c(G, c, S, \mathcal{R}) = 1$. Then

$$\max_c \frac{T_c(G, c, S, \mathcal{R}^*)}{T_f^{av}} \ge \frac{1}{H_N} \max_c \frac{T_c(G, c, S, \mathcal{R})}{T_f},$$

giving

$$\max_c T_f^{av}(\mathcal{R}^*) \le H_N \max_c T_f.$$

Note that the maximum value of T_f and T_f^{av} is not necessarily achieved for the same capacity vector c, or for the same number of receivers N. What this theorem tells us is that, for a given $\{G, S, \mathcal{R}\}$, with $|\mathcal{R}| = N$, the maximum common rate we can guarantee to all receivers will be at most H_N times smaller than the maximum average rate we can send from S to any subset of the receivers $\mathcal{R}$. The theorem quantitatively bounds the advantage in going from the stricter measure $\alpha(G, S, \mathcal{R})$ to the weaker measure $\beta(G, S, \mathcal{R}^*)$. Furthermore, it is often the case that for particular instances of $(G, S, \mathcal{R})$, either $\alpha(G, S, \mathcal{R})$ or $\beta(G, S, \mathcal{R}^*)$ is easier to analyze and the theorem can be useful to get an estimate of the other quantity.

We comment on the tightness of the bounds in the theorem. There are instances in which $\beta(G, S, \mathcal{R}^*) = 1$, take for example the case when G is a tree rooted at S. On the other hand there are instances in which $\beta(G, S, \mathcal{R}^*) = O(1/\ln N)\alpha(G, S, \mathcal{R})$. Examples include bipartite instances discussed in the next section and also instances defined in [8]. In general, the ratio $\alpha(G, S, \mathcal{R})/\beta(G, S, \mathcal{R}^*)$ can take on a value in the range $[1, H_N]$.

IV. Configurations with Small Network Coding Benefits

We here describe classes of networks for which network coding can at most double the sum rate achievable by routing. Note that at no example in this section do we talk about optimal routing that is in general NP-hard, but only about simple routing schemes which achieve a certain fraction of the coding throughput.

A. *Configurations with Two Receivers*

Consider the case of an arbitrary network with h sources and $N = 2$ receivers R_1 and R_2. The throughput achievable by network coding is $T_c = h = 2$. In the scenario when only receiver R_1 uses the network, no coding is required, and the throughput to R_1 is h. Therefore, we have

$$\frac{1}{2} \leq \frac{T_i^{av}}{T_c} \leq 1.$$

B. *Configurations with Two Sources*

For network with two sources, the bounds in (1) give

$$\frac{1}{2} \leq \frac{T_i^{av}}{T_c} \leq 1$$

by setting $h = 2$. We can tighten the lower bound as follows:

Theorem 2: For all networks with $h = 2$ sources and N receivers, if the min-cut condition is satisfied for every receiver, it holds that

$$\frac{T_i^{av}}{T_c} \geq \frac{1}{2} + \frac{1}{2N}.$$

There are networks for which the bound holds with equality.

Proof: Consider a minimal information flow graph, and choose one of the sources to transmit to all the coding points in the information flow graph. Since the configuration is minimal, the other source node contains at least one receiver ([16] Theorem 3). Therefore, at least one of the receivers will be able to receive both sources. Thus a lower bound on the achievable T_i^{av} throughput is $\frac{N+1}{N}$; hence the bound in the theorem.

Note that the bound is achievable, since for every N, there exist minimal configurations where without network coding we can not achieve sum throughput better than $N+1$. Such configurations are the minimal information flow graphs with $N-1$ coding points, described in ([16] Theorem 4). For these configurations, each of the two source nodes contains one receiver node, thus we immediately start with sum rate 2. Moreover each of the $N-1$ coding points contains exactly two receiver nodes. Using routing, only one of the two receiver nodes in each coding point will collect incremental information. This fact can be proved by using induction on the number of coding nodes and the fact that such a minimal configuration with N coding nodes can be created by a minimal configuration with $N-1$ coding points by adding one receiver. Thus we can achieve sum rate $2+N-1=N+1$ and $T_i^{av}=\frac{1}{N}+1$. ■

There are networks with two sources with even smaller coding throughput advantage. Consider, for example, the network in Fig. 2. Two sources are connected through $q+1$ intermediate nodes

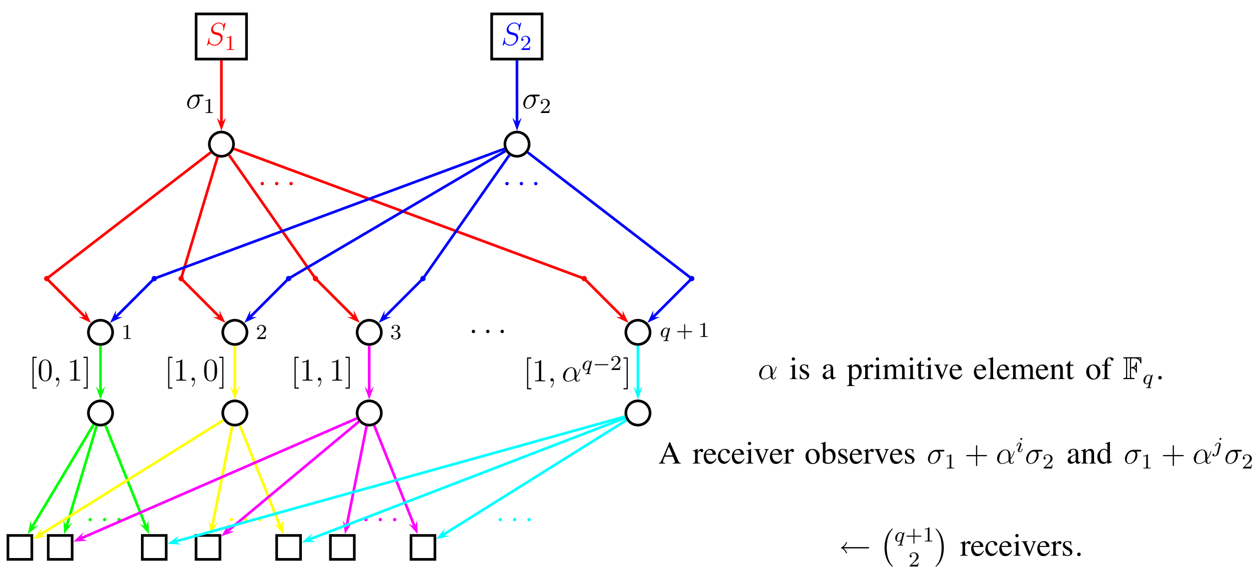


Fig. 2. A network with two sources and $\binom{q+1}{2}$ receivers.

and branches to $\binom{q+1}{2}$ receivers. The network code which achieves the $T_c=2$ is also explained in the figure. Note that the alphabet size required to achieve this throughput equals q. We show that with routing we can achieve the average throughput of at least $\frac{3}{4}T_c$. We route S_1 through one half of the $q+1$ intermediate nodes, and S_2 through the other half. Therefore, the average routing throughput, for even $q+1$, is given by

$$T_i^{av} = \frac{1}{\binom{q+1}{2}}\left[\frac{q+1}{2}\left(\frac{q+1}{2}-1\right)\cdot 1+\left(\frac{q+1}{2}\right)^2\cdot 2\right] > \frac{3}{4}\cdot T_c.$$

Note that the routing throughput does not depend on q. Thus routing may be of interest when the number of receivers is large and consequently coding requires a large alphabet size.

C. Bipartite Configurations with 2–Input Coding Points

Proposition 1: Consider a bipartite information flow graph with h sources and N receivers. Assume that each coding point has two parents which are source nodes. Then

$$\frac{T_i^{av}}{T_c} \geq \frac{1}{2}. \tag{8}$$

Proof: Since each coding point c has two parents, connecting it to sources $S_1(c)$ and $S_2(c)$, it contains $N_1 \geq 1$ receivers observing source $S_1(c)$ and $N_2 \geq 1$ receiver observing source $S_2(c)$. If $N_1 \geq N_2$, we assign to c source $S_1(c)$, and source $S_2(c)$ otherwise. This way we ensure that by merely routing at each coding point, at least half of its receivers observe one of its inputs. Note that a receiver is observing a particular source at exactly one coding point. Therefore the total routing throughput is at least half of the total throughput achievable by coding. ■

D. Configurations with h-input Coding Points

We first consider networks with h sources and N receivers whose minimal information flow graphs are bipartite and each coding point has h inputs. An example of such networks is illustrated in Fig. 3. In network coding literature, these networks are known as combination networks $B(h,k)$. There are three layers of nodes. The first layer contains the source node, at which h information sources are available. The second layer contains kh nodes connected to the source node. The third layer contains $\binom{kh}{h}$ receiver nodes. Note that each h nodes of the second layer are observed by a receiver. This example was introduces in [5] to illustrate the benefits of network coding in terms of the integral throughput T_i. We look into the average throughput benefits first.

Theorem 3: The average throughput benefits of network coding for combination networks $B(h,k)$ is bounded as

$$\frac{T_i^{av}}{T_c} > 1-\frac{1}{e}, \tag{9}$$

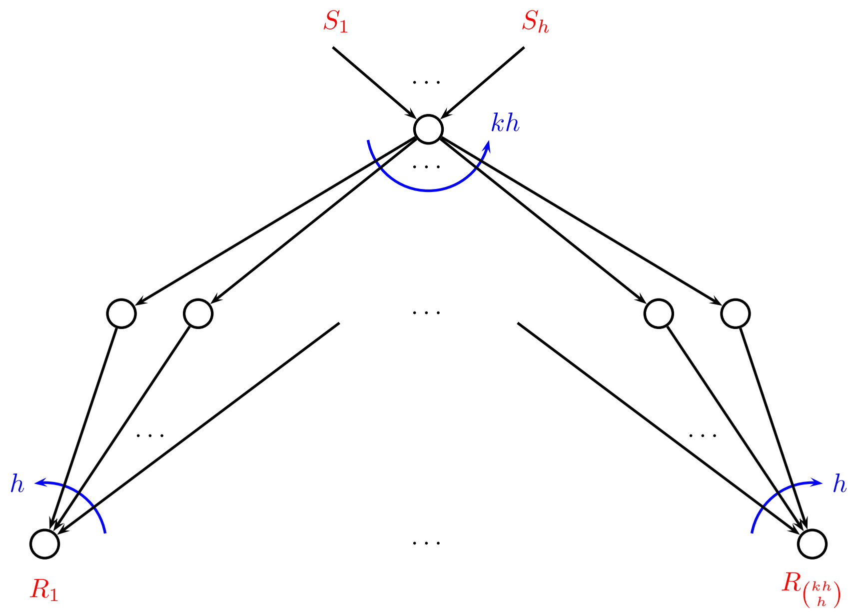


Fig. 3. Combination $B(h, k)$ network.

for all h and k.

Proof: Note that the min-cut condition is satisfied for every receiver, and thus $T_c = h$. Route each of the sources through exactly k edges going out of the source node. Let M_i denote the number of receivers that do not receive source S_i, under this routing scheme. The total loss of throughput will be equal to $\sum_{i=1}^{h} M_i$. Since source S_i is transmitted to k nodes, there exist $M_i = \binom{kh-k}{h}$ receivers that do not receive source S_i. Using symmetry, the total loss in throughput is $h\binom{kh-k}{h}$ and thus

$$T_i^{av} = \left[h\binom{kh}{h} - h\binom{kh-k}{h} \right] / \binom{kh}{h}.$$

The ratio between the routing and coding throughput can, therefore, be lower-bounded as

$$\begin{aligned}\frac{T_i^{av}}{T_c} &= \frac{h\binom{kh}{h} - h\binom{kh-k}{h}}{h\binom{kh}{h}} \\ &= 1 - \frac{\binom{kh-k}{h}}{\binom{kh}{h}} \\ &= 1 - \prod_{i=0}^{h-1}\Big(1 - \frac{k}{kh-i}\Big) \\ &> \Big(1 - \frac{1}{h}\Big)^h > 1 - \frac{1}{e}.\end{aligned}$$

■

However, the benefits of network coding as compared to the fractional and integral routing are much higher. It straightforward to upper-bound the fractional throughput of combination networks $B(k,h)$. Note that each Steiner tree needs $kh-(h-1)$ out of the kh edges going out of the source node. Therefore, the fractional packing number is at most $kh/(kh-h+1)$, and consequently

$$\frac{T_f}{T_c} \leq \frac{k}{h(k-1)+1}. \tag{10}$$

The above bound is a special case of the result obtained in [13]. The network coding benefits of integral routing can be bounded as

$$\frac{T_i}{T_c} \leq \frac{1}{h},$$

since we can only have exactly one Steiner tree. Note that for the $B(h,k)$ networks, $h \approx lnN$, and the bound in Theorem 1 is tight. Indeed, comparing Eq. (9) and (10) we get that

$$\frac{T_f}{T_c} = O(1/\ln N)\frac{T_i^{av}}{T_c} = O(1/\ln N)\frac{T_f^{av}}{T_c}.$$

In Sec. IV-E, we will show a way to make the integral routing throughput T_i equal to the average by the employing a suitable erasure correcting code.

We now examine more general configurations. The following theorem removes the bipartite graph assumption.

Theorem 4: Consider a information flow configuration with h sources and N receivers. Assume that the vertex min-cut to each coding point is h, and that each subset of h coding points shares a receiver. Then

$$\frac{T_i^{av}}{T_c} \geq 1 - \frac{1}{e}. \tag{11}$$

Proof: Assume that the number of coding points is kh. It is sufficient to show that we can route each source to k coding points, since the claim then follows from the result of Thm. 3. In other words, it sufficient to show that our graph can be decomposed into to h vertex-disjoint trees, each tree rooted at a different source node, since then we can route each source to its corresponding tree.

Let $\tau_i = (V_i, E_i)$ denote the tree through which we will route source S_i. We will first create τ_1, then τ_2, and continue to τ_h. Consider source S_1. We are going to construct τ_1 in k steps, where in each step we will add one vertex and one edge to τ_1. Let V_1^i and E_1^i denote the vertices and edges respectively that are allocated to τ_1 at step i. Initially $V_1^1 = \{S_1\}$, where with S_1 we denote the node corresponding to source S_1, and $E_1 = \{\}$. At step i, we add a coding point C_i to the set V_1^i that has a parent P_i in V_1^i, to create $V_1^{i+1} = \{V_1^i \bigcup C_i\}$ and $E_1^{i+1} = \{E_1^i \bigcup (P_i, C_i)\}$. We then remove all incoming edges to C_i, apart from (P_i, C_i). We want to choose a C_i so that after removing these edges the vertex min-cut property towards the rest of the coding points is not affected. That is, for the rest of the coding points, there still exists h vertex disjoint paths, one that starts from any vertex of V_1^{i+1} and $h-1$ that start from the source nodes $S_2 \dots S_h$. It is sufficient to show that such a C_i always exists.

From the theorem assumption, each coding point has h parents $P_1, \dots P_h$. Any operation in the graph that does not affect the min-cut property of $P_1, \dots P_h$ will not affect the min-cut property of their child either. Thus, if we add coding point C_i to the set V_1^i, we need to make sure that the min-cut property is not violated *only* for the coding points that have a parent in the set $\{V_1^i \bigcup C_i\}$. Assume that adding C_i to V_1^i violates the min-cut property for some coding point C_j. Then C_j is a child of C_i and another node $P_j \in V_i$. To see that, note the following:

1) If a set of nodes is affected, at least one of them, say C_j, is a child of C_i.
2) If C_j is a child of $P_1 = C_i$ and none of its remaining $h-1$ parents $P_2, \dots P_h$ belongs in V_1^i, because the mincut to $P_2, \dots P_h$ is h, allocating source S_1 to $P_1 = C_i$ cannot affect the min-cut condition, C_j can still receive the remaining $h-1$ sources through $P_2, \dots P_h$. Thus, if the C_j's min-cut condition is violated, C_j must have another parent, say P_j, in V_i.

We can then examine whether we can add C_j to the set V_1^{i+1}, i.e., whether it is possible to have $V_1^{i+1} = \{V_1^i \bigcup C_j\}$ and $E_1^{i+1} = \{E_1^i \bigcup (P_j, C_j)\}$. This will not be possible only if C_j also has a child in common with a parent node in V_1^i. I We can repeat this procedure following such parent-child relations, until we find a set V_1^{i+1} that does not violate the min-cut condition. Since the graph is finite, this procedure will identify a coding point that is a child of a vertex in V_i and does not have any child in common with any vertex in V_i.

Following this procedure, we can create a tree τ_1 that contains k subtrees. We then remove τ_1 from the information flow graph, and all the edges adjacent to vertices in τ_1. We are now left with an information flow graph with $h-1$ sources such that the min-cut to each coding point is $h-1$, and we can repeat the same procedure. ∎

We next examine the case of a bipartite graph where every coding point has h parents, but no constraint is placed on how the receivers are distributed. Combination networks as shown in Fig. 3, but with arbitrary number of receivers, belong to this class of networks.

Theorem 5: Consider a bipartite information flow configuration with h sources and N receivers. Assume that each coding point has h parents, and that allocation of the sources to the coding points is done uniformly at random. Then, each receiver will on the average experience the integral throughput T_i^{av} satisfying

$$\frac{T_i^{av}}{T_c} \geq 1 - \frac{1}{e}. \tag{12}$$

Proof: For each receiver, this scenario is a classic occupancy model in which h balls, corresponding to the receiver's h leaves (incoming edges) are thrown independently and uniformly into h urns corresponding to the h sources. Let T_i be the random variable representing the number of occupied bins (sources a receiver observes). Then, for this occupancy model, we have (see for example [19, Ch. 1])

$$T_i^{av} = h\Big[1 - \Big(1 - \frac{1}{h}\Big)^h\Big]. \tag{13}$$

Therefore, the ratio between the expected throughput when no coding is used and the average throughput when coding is used is given by

$$\frac{T_i^{av}}{T_c} \geq \Big[1 - \Big(1 - \frac{1}{h}\Big)^h\Big] > 1 - \frac{1}{e}.$$

∎

In the combination network example in Fig. 3, this corresponds to the routing strategy in which the source to be routed through an edge going out of the source node is chosen uniformly at random from the h information sources.

The connection with the classic occupancy model enables us to directly obtain several other results listed below. The results can be easily derived from the material in [19, Ch. 1].

Theorem 6: For each receiver, the probability distribution of the random variable T_i representing the number of observed sources (filled urns) is given by

$$\Pr\{T_i = k\} = \binom{h}{k}\Big(1 - \frac{h-k}{h}\Big)^h \Pr\{\mu_0(k) = 0\}$$

$$\text{where } \Pr\{\mu_0(k) = 0\} = \sum_{l=0}^{k} \binom{k}{l}(-1)^l\Big(1 - \frac{l}{k}\Big)^h.$$

Theorem 7: As $h \to \infty$, the mean and the variance of T_i behave as follows:

$$T_i^{av} \to h(1 - (1 - e^{-1}) \text{ and } \sigma^2(T_i) \to h(1 - e^{-1})(1 - 2e^{-1}).$$

Theorem 8: As $h \to \infty$, the probability that the observed throughput T_i is different from its becomes exponentially small:

$$\Pr\left\{\frac{T_i - T_i^{av}}{\sigma(T_i)} < x\right\} \to \frac{1}{2\pi}\int_{-\infty}^{x} e^{-u^2/2}du < e^{-x^2/2}.$$

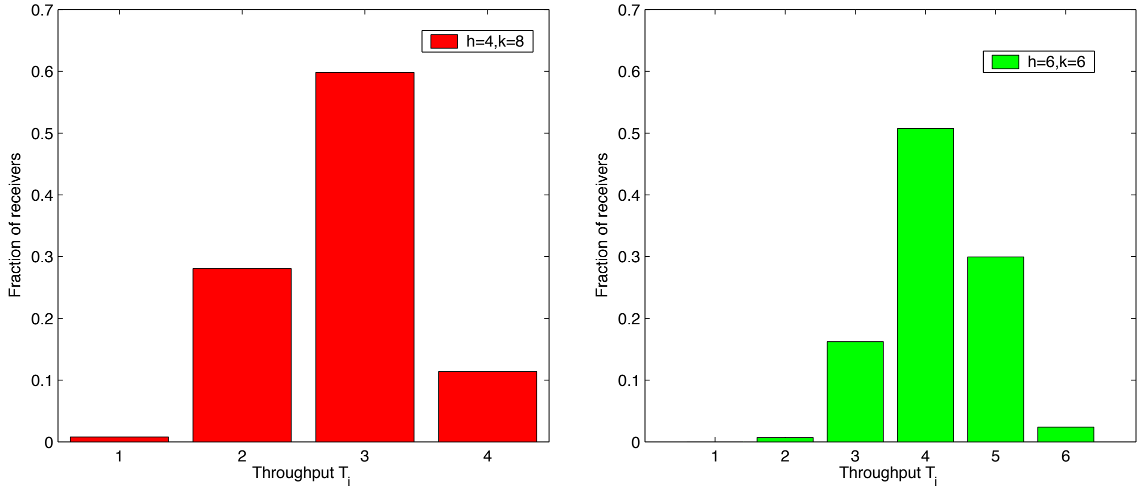
Theorem 8 makes the point that looking at the average throughput is a reasonable choice. This is especially true when the number of receivers is large, and the throughput they experience tends to concentrate around a much larger value than the minimum. For example, Fig. 4 plots how the throughput is distributed among the receivers for two bipartite configurations $B(h, k)$. In both cases the fraction of receivers that observe throughput $T_i = 1$ is very small as compared to the number of receivers that experience throughput $T_i \geq h/2$.

In the following we first describe a joint routing-coding scheme that achieves $T_i = T_i^{av}$ asymptotically for the set of configurations $B(h, k)$ and then discuss how this scheme can be possibly generalized to arbitrary configurations.

E. Achieving the Average Throughput for all Receivers by Channel Coding

We here introduce time as an additional dimension in our routing problem, which is in network coding literature known as *vector routing* [13]. We show that by combined vector routing and channel coding, the integral throughput can achieve the average asymptotically over time.

Consider the combination networks as shown in Fig. 3 but with arbitrary number of receivers, where the information source to be routed through an edge going out of the source node is



(a) h=4 k=8 (b) h=6 k=6

Fig. 4. Histogram depicting on the y-axis the normalized number of receivers and on the x-axis the throughput T_i the receivers experience with routing, for two bipartite multicast configurations with h sources and kh subtrees.

chosen uniformly at random from the h information sources. The probability that a receiver will not observe source S_i is given by

$$\epsilon = \left(\frac{h-1}{h}\right)^h. \tag{14}$$

Therefore, with this routing strategy, the expected value of the integral throughput is given by

$$T_i^{av} = h\Big[1 - \left(\frac{h-1}{h}\right)^h\Big] = h(1-\epsilon). \tag{15}$$

Recall that we have obtained this result in Theorem 5, together with the entire probability distribution for the random variable T_i in Sec. IV-C.

We can apply this random coding scheme over n time-slots where at each time slot an independent experiment takes place. That is, each of the h information sources produces a sequence of symbols of length n. Routing is performed in the described random manner at each time slot, when a set of h symbols is available for transmission by the h sources. Note that for n large enough, each receiver j will observe over time throughput $E(T_i^j) = T_i^{av}$.

Under this scenario, a receiver observes the sequence of each source outputs as if it had passed through an erasure channel with the probability of erasure ϵ given by (14). Therefore, the symbols of each source can be encoded by an erasure-correcting code of rate k/n which will allow recovering the k information symbols after n transmissions.

Theorem 9: For the combination networks as shown in Fig. 3 but with arbitrary number of receivers, there exist a sequence of channel codes of rates $k/n \to 1-\epsilon$ and a routing strategy such that the integral throughput $T_i(n) \to hk/n \to T_i^{av}$ as $n \to \infty$.

Proof: Under the routing strategy described above, a receiver observes the sequence of each source outputs as if it had passed through an erasure channel with the probability of erasure ϵ given by (14). The channel capacity of such a channel is equal to $1-\epsilon$, and there exists a sequence of codes with rates $k/n < 1-\epsilon$ such that the probability of incorrect decoding goes to 0 as $n \to \infty$. Therefore, since there are h sources, we have $T_i(n) \to h \cdot k/n$ as $n \to \infty$. Since k/n can be taken arbitrary close to the capacity, we have $T_i(n) \to h(1-\epsilon) = T_i$, where the last equality follows from (15). ■

The result immediately generalizes to any bipartite information flow configuration with h sources where each coding subtree has h parents and allocation of the sources to the coding subtrees is done uniformly at random. When the configuration is symmetric, as in the case of $B(h,k)$ networks, the random routing can be replaced by deterministic, and the integral throughput T_i can achieve the average after a finite number of time units. For example, in the case of $B(h,k)$ networks, the routing strategy can circulate over the $n \triangleq (kh)!/(k!)^h$ possible assignments of h sources to kh edges s.t. each source is assigned to exactly k edges. After a sequence of length n is transmitted from each source, a receiver will have exactly

$$n - m \triangleq \binom{kh-h}{k}\frac{(kh-k)!}{(k!)^{h-1}}$$

symbols erased from each source. Thus the fraction of received symbols per source is given by

$$1 - \frac{\binom{kh-h}{k}\frac{(kh-k)!}{(k!)^{h-1}}}{\frac{(kh)!}{(k!)^h}} = 1 - \frac{\binom{kh-k}{h}}{\binom{kh}{h}} = \frac{T_i^{av}}{h}.$$

Therefore, employing an (n,m) Reed-Solomon code at each source would result in $T_i(n) = T_{av}$.

Note that this scheme cannot be implemented with fractional (scalar) routing. One way to understand this, is to describe our integral routing scheme at a given time slot with an $1 \times kh$ vector from an alphabet of size h. The ith element of the vector expresses what source is routed through the the ith edge going out of the source node. By the weak law of large numbers, if we uniformly at random choose the elements of a vector from an alphabet of size h, as we do for our routing scheme, we will effectively select with equal probability one of the *typical vectors* of this random experiment, i.e., a vector where asymptotically each of the alphabet symbols will appear an equal number of times. Thus, averaging over n time slots, we are averaging over the set of typical vectors of size $1 \times kh$. Fractional routing amounts to averaging over the set of all $1 \times kh$ vectors, not only the typical ones (an a-typical vector is for example to get the first source allocated to all kh edges).

We now formulate the problem of creating an appropriate routing schedule that supports such a coding scheme as a linear program. We adopt the notation of Section III and consider an instance $\{G, S, \mathcal{R}\}$. Let τ denote the set of partial Steiner trees in G rooted at S with terminal set $\mathcal{R}$. We will be using the number of time slots, n as a parameter. In each time slot we seek a feasible fractional packing of partial Steiner trees. The goal is to maximize the total number of trees that each receiver occurs in, across the time slots. We express this as a linear program as follows. For a tree $t \in \tau$ and a time slot k we have a non-negative variable $y(t,k)$ that indicates the amount of t that is packed in time slot k.

$$\max f$$

$$\begin{aligned} \sum_k \sum_{t\in\tau : R_i \in t} y(t,k) &\geq f, \quad \forall \; R_i \\ \sum_{t\in\tau : e\in t} y(t,k) &\leq c_e, \quad \forall \; e \in E, 1 \leq k \leq n \\ y(t,k) &\geq 0, \quad \forall \; t \in \tau, 1 \leq k \leq n \end{aligned}$$

Given a solution y^* to the above linear program, let $z_i = \sum_k \sum_{t:R_i\in t} y^*(t,k)$ be the total amount that R_i appears in the packing. Let $m = \sum_{i=1}^N z_i$. We can use an erasure code that employs m coded symbols to convey the same f information symbols to all receivers, i.e., achieves rate $T_i = \frac{f}{m}$. (To be precise, we need $\frac{f}{m}$ to be a rational and not a real number.) For integer edge capacities c_e, there is an optimum solution with rational coordinates.

The described scheme can be viewed as a generalization of the vector routing solution described in [13]. The vector routing solution in [13], similarly to our approach, uses time as an additional dimension. The difference is that in [13] we are still trying to find Steiner trees that span all receivers (albeit not necessarily at the same time-slot), that is, perform packing of Steiner trees in G'. In our scheme, we allow the flexibility of packing partial Steiner trees, thus possibly achieving a higher rate, and then use an erasure correcting code to convey common information. Also note that our scheme does not employ coding at intermediate nodes, only at the source nodes. Thus, it offers an upper bound on the maximum throughput we may achieve without allowing intermediate nodes in the network to code, *i.e.*, without using network coding.

V. CONFIGURATIONS WITH LARGE NETWORK CODING BENEFITS

We here describe a class of networks for which network coding can offer up to $\sqrt{N}$-fold increase of the average throughput achievable by routing. This class of networks, which we call $ZK(p,N)$, was originally described by Zosin and Khuller in [7] to demonstrate the integrality gap of standard LP for the directed Steiner tree problem.

A. The Network $ZK(p,N)$

Let N and p, $p \le N$, be two integers and $\mathcal{I} = \{1, 2, \ldots, N\}$ be an index set. We define two more index sets: $\mathcal{A}$ as the set of all $(p-1)$-element subsets of $\mathcal{I}$ and $\mathcal{B}$ as the set of all p-element subsets of $\mathcal{I}$. We consider a class of layered acyclic networks $ZK(p,N)$, illustrated in Fig. 5, and defined by the two parameters N and p as follows: Source S transmits information

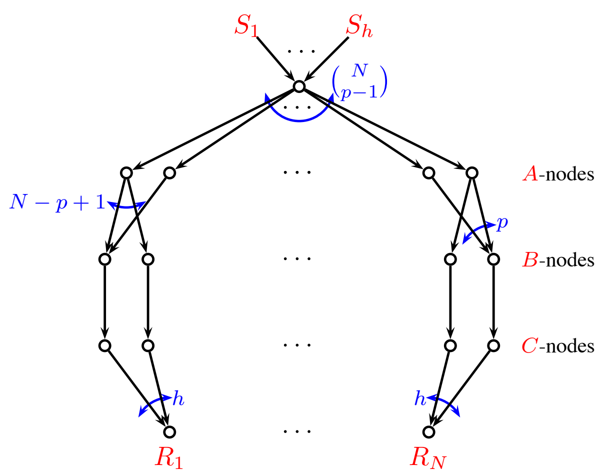


Fig. 5. The network configuration $ZK(p,N)$. The min-cut to each of the N receivers is $h = \binom{N-1}{p-1}$.

to N receiver nodes $R_1 \ldots R_N$ through a network of three sets of nodes A, B and C. A-nodes are indexed by the elements of $\mathcal{A}$, and B and C-nodes, by the elements of $\mathcal{B}$. An A node is connected to a B node if the index of A is a subset of the index of B. A B node is connected to a C node if and only if their indices are identical. A receiver node is connected to the C nodes whose indices contain the index of the receiver. All edges in the graph have unit capacity. The out-degree of the source node is $\binom{N}{p-1}$. Two specific members of this family of networks are shown in Fig. 6 and Fig. 7.

We can compute the degrees of the nodes in the network by simple combinatorics:

Proposition 2:

- the out-degree of A nodes is $N-(p-1)$,
- the in-degree of B nodes is p,
- the out-degree of C nodes is p,
- the in-degree of the receiver nodes is $\binom{N-1}{p-1}$.

We next compute the value of the min-cut between the source node and each receiver node, or equivalently, the number of edge disjoint paths between the source and each receiver.

Theorem 10: There are exactly $\binom{N-1}{p-1}$ edge disjoint paths between the source and each receiver.

Proof: Consider receiver i. It is connected to the $\binom{N-1}{p-1}$ distinct C-nodes indexed by the elements of $\mathcal{B}$ containing i. Each of the C-nodes is is connected to the B-node with the same index. All paths between the source and the receiver i have to go through these B and C-nodes. Therefore the number of edge disjoint paths between the source and the receiver can not be larger than $\binom{N-1}{p-1}$. To show that there exist that many edge disjoint paths, we proceed as follows: After removing i from the indices of the B-nodes receiver i is connected to, we are left with $\binom{N-1}{p-1}$ distinct sets of size $p-1$, *i.e.* distinct elements of $\mathcal{A}$. We use the A-nodes indexed by these elements of $\mathcal{A}$ to connect the receiver i B-nodes to the source. ■

Therefore, the sum rate with network coding NT_c is equal to $N\binom{N-1}{p-1}$. We next find an upper bound to the sum rate without network coding T_f and to the ratio $T_{f^{av}}/T_c$.

Theorem 11: In a network in Fig. 5 where $h=\binom{N-1}{p-1}$,

$$\frac{T_f^{av}}{T_c} \leq \frac{p-1}{N-p+1} + \frac{1}{p}. \tag{16}$$

Proof: If only routing is permitted, the information is transmitted from the source node to the receiver through a number of trees, each carrying a different information source. Let a_t be the number of A-nodes in tree t, and c_t, the number of B and C-nodes. Note that $b_t \geq a_t$, and that the c_t C-nodes are all descendants of the a_t A-nodes. Therefore, we can count the number of the receivers spanned by the tree as follows: Let $n_t(A(j))$ be the number of C-nodes connected to the jth A-node in the tree. Note that

$$\sum_{j=1}^{a_t} n_t(A(j)) = c_t.$$

The maximum number of receivers the tree can reach through this A-node is $n_t(A(j)) + p - 1$. Consequently, the maximum number of receivers the tree can reach is

$$\sum_{j=1}^{a_t} [n_t(A(j)) + p - 1] = a_t(p-1) + c_t.$$

To find and upper bound to the routing throughput, we need to find the number of receivers that can be reached by a set of disjoint trees. Note that for any set of disjoint trees we have

$$\sum_t a_t \leq \binom{N}{p-1} \text{ and } \sum_t c_t \leq \binom{N}{p}.$$

Therefore, T_u can be upper-bounded as

$$\begin{aligned} T_i &\leq \frac{1}{N} \sum_t (a_t(p-1) + c_t) \\ &= \frac{1}{N}(p-1) \sum_t a_t + \sum_t c_t \leq (p-1)\binom{N}{p-1} + \binom{N}{p}. \end{aligned} \tag{17}$$

The sum rate with network coding T_c is equal to $N\binom{N-1}{p-1}$. Thus we get that

$$\frac{T_i^{av}}{T_c} \leq \frac{p-1}{N-p+1} + \frac{1}{p}.$$

We can apply the exact same arguments to upper bound T_f^{av}, by allowing a_t and c_t to take fractional values, and interpreting these values as the fractional rate of the corresponding trees.

■

For a fixed N, the LHS of the above inequality is minimized for

$$p = \frac{N+1}{\sqrt{N}+1} \approxeq \sqrt{N},$$

and for this value of p,

$$\frac{T_f^{av}}{T_c} \leq 2\frac{\sqrt{N}}{1+N} \lessapprox \frac{2}{\sqrt{N}}. \tag{18}$$

B. Deterministic Coding

We show that for the $ZK(p, N)$ configurations there exist network codes over the binary alphabet. Thus, very simple operations are sufficient to achieve significant throughput benefits. We first explain how the coding is done for two special cases of p: when $p = 2$ and when $p = N - 1$, and then proceed with the general case.

1) $p = 2$: Consider the case $ZK(2, N)$ where $p = 2$ and N is arbitrary. An example for $N = 4$ is shown in Fig. 6. In this case the number of information sources is $h = N - 1$. We can

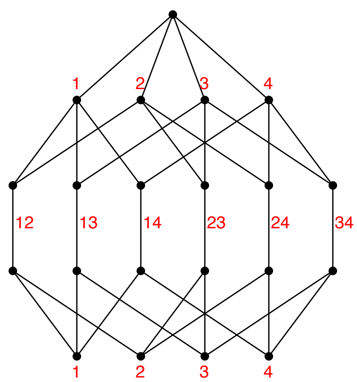


Fig. 6. The network $ZK(p = 2, N = 4)$.

code over the binary field as follows: Since the number of edges going out of S into A nodes is N, we can send the $N - 1$ sources over the first $N - 1$ of this edges and not use the Nth edge. In other words, the coding vector of the ith of this edges is the ith basis vector e_i for $i = 1, 2, \ldots, N - 1$. The B-nodes merely sum their inputs over $\mathbb{F}_2^h$, and forward the result to the C-nodes. Consequently, the coding vectors on the branches going to receiver N are the $N - 1$ basis vectors, and the coding vectors on the branches going to receiver i for $i = 1, 2, \ldots, N - 1$ are e_i and $e_j + e_i$ for $j = 1, \ldots, N - 1$ and $j \neq i$.

2) $p = N - 1$: Consider the case when $p = 2$ for arbitrary N. An example for $N = 5$ is shown in Fig. 7. In this case the number of information sources is $h = N - 1$. The number of C-nodes is N. Each subset of $N - 1$ C-nodes is observed by a receiver. Therefore, any $N - 1$ of coding vectors of the edges between the B and C-nodes should be linearly independent. The

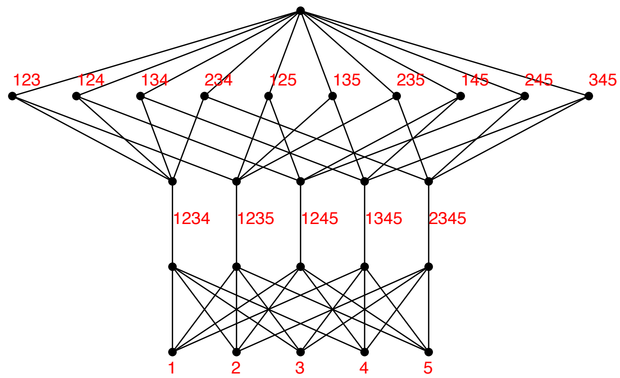


Fig. 7. The network $ZK(p = 4, N = 5)$.

following list of vectors can be used for coding along theses edges:

$$\begin{array}{cccc} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 1 \\ 1 & 1 & \dots & 1 \end{array} \tag{19}$$

We can obtain this edge by coding as follows: To the $N-1$ edges going from the source to the A nodes whose label does not contain N, we assign $N-1$ basis vectors of over $\mathbb{F}_2^{(N-1)}$. We remove all other edges outgoing of the source, and then all A-nodes which lost their connection with the source, and the edges coming out of the removed A nodes. Consequently, the first of the B-nodes has $N-1$ inputs. By addition, of these inputs the coding vector between this B and its corresponding C node becomes $[1\,1\,...\,1]$. The rest of the B-nodes have only one input. Thus we get the binary edge 19) at the last set of edges.

3) The General Case: For arbitrary values of p and N, network coding can be done as follows: We first remove the edges going out of S into those A-nodes whose labels contain N. There are $\binom{N-1}{p-2}$ such edges. Since the number of edges going out of S into A-nodes is $\binom{N}{p-1}$, the number of remaining edges is $\binom{N}{p-1} - \binom{N-1}{p-2} = \binom{N-1}{p-1}$. We label these edges by the $h = \binom{N-1}{p-1}$ different basis elements of $\mathbb{F}_2^h$. We further remove all A-nodes which have lost their connection with the source S, as well as their outgoing edges. The B-nodes merely sum their inputs over $\mathbb{F}_2^h$, and forward the result to the C-nodes.

Consider a C-node that the Nth receiver is connected to. Its label, say ω, is a p-element subset of $\mathcal{I}$ containing N. Because of of our edge removal, the only A-node that this C-node is connected to is the one with the label $\omega \setminus \{N\}$. Therefore, all C-nodes that the Nth receiver is connected to have a single input, and all those inputs are different. Consequently, the Nth receiver observes all the sources directly.

Each of the receivers $1, 2, \ldots, N-1$ will have to solve a system of equations. Consider one of these receivers, say j. Some of the C-nodes that the jth receiver is connected to have a single input: those are the nodes whose label contains N. There are $\binom{N-2}{p-2}$ such nodes, and they all have different labels. For the rest of the proof, it is important to note that each of these labels contains j, and the $\binom{N-2}{p-2}$ labels are all $(p-1)$-element subsets of $\mathcal{I}$ which contain j and do not contain N. Let us now consider the remaining $\binom{N-1}{p-1} - \binom{N-2}{p-2} = \binom{N-2}{p-1}$ C-nodes that the jth receiver is connected to. Each of these nodes is connected to p A-nodes. The labels of $p-1$ of these A-nodes contain j, and only one does not. That label is different for all C-nodes that the receiver j is connected to. Consequently, the jth receiver gets $\binom{N-2}{p-2}$ sources directly, and each source of the remaining $\binom{N-2}{p-1}$ as a sum of that source and some $p-1$ of the sources received directly.

C. *Random Coding*

1) General Networks: For a general network with N receivers in which coding is performed by random assignment of coding vectors over the alphabet $\mathbb{F}_q$, a lower bound to the probability P_N^d that all N receivers will be able to decode is derived in [15] to be

$$P_N^d \geq \left(1 - \frac{N}{q}\right)^n,$$

where n is defined in [15] to be the number of edges where coding is performed. For the $ZK(p, N)$ configurations, $n \geq \binom{N}{p}$, and the lower bound becomes

$$P_N^d \geq \left(1 - \frac{N}{q}\right)^{\binom{N}{p}} \approxeq e^{-\frac{N\binom{N}{p}}{q}}.$$

We next derive randomized coding bounds that apply specifically to the $ZK(p, N)$ configurations. We first consider the case when randomized coding is used at all nodes with multiple inputs, namely the source node and all the B nodes, and then the case when the coding at the source node is done deterministically as in Sec. V-B.3, and randomized coding is done at the B nodes with multiple inputs after the removal of edges as in Sec. V-B.3.

2) Random Coding for the Special Class of Networks: First, randomized coding is used at the source node to decide which linear combination goes to each A-node. Then:

Pr (receiver j has a full rank set of equations)=

Pr (each node C receiver j observes increases his rank)=

$\Pi_{i=2}^{h}$ Pr (node C_i that receiver j observes increases his rank)=

Π Pr (node C_i increases receiver j rank$|\{A_i\}$ inputs of C_i do not lie in the span of $\{C_1 \dots C_{i-1}\}$)Pr($\{A_i\}$ inputs of C_i do not lie in the span of $\{C_1 \dots C_{i-1}\}$) =

$\geq \Pi(1-\frac{1}{q})^2 = (1-\frac{1}{q})^{2(h-1)} = (1-\frac{1}{q})^{2(\binom{N-1}{p-1}-1)}$.

3) Random Coding at B nodes: Assume that we choose the coding vectors for the edges going into the A-nodes as we did for the deterministic coding described in Sec. V-B.3, but now the B-nodes randomly combine their inputs instead of summing them.

Consider receiver j. As before $\binom{N-2}{p-2}$ or its C-nodes are connected to a single input. Consider one of the remaining $\binom{N-2}{p-1}$ C-nodes that the receiver j is connected to. The corresponding B node will form a random linear combination of the $p-1$ sources that are directly received and of an additional source. Therefore, if the random linear combining is performed over $\mathbb{F}_q$, the C will observe a linear combination of only the $p-1$ sources directly received with probability $1/q$, namely only if the coefficient zero is chosen for the additional source. Thus the receiver j receives an independent linear combination from a C node with p inputs with probability $1-1/q$. Since the linear combining at each multi-input B node is performed independently, receiver j will be able to decode all h sources with probability

$$\Pr\{\text{single receiver decodes}\} = \left(1-\frac{1}{q}\right)^{\binom{N-2}{p-1}}.$$

We can also compute the probability that all receivers be able to decode all sources. Note that this happens when all multi-input B nodes use a nonzero coefficient for the not-directly received source. Since there are $\binom{N}{p}-\binom{N-1}{p-1}=\binom{N}{p-1}$ such nodes, we obtain

$$\Pr\{\text{all receiver decode}\} = \left(1-\frac{1}{q}\right)^{\binom{N}{p-1}}.$$

Thus similarly with before, if we want this probability to be greater than e^{-1}, we need to choose

$q \geq \binom{N}{p-1}$. We conclude that randomized coding might require an exponentially larger alphabet size over the network configurations $ZK(p, N)$.

D. The Information Flow Graph Properties

The coding scheme described in Section V-B.3 removed edges from the source to the A-nodes, and thus effectively transformed the configuration to a bipartite one (the corresponding information flow graph is bipartite). In Section IV we saw that for a number of families of bipartite graphs, the average throughput is comparable to the network coding throughput. In this section we examine in more detail the structure of the $ZK(p, N)$ configurations and discuss a number of interesting properties.

We start by describing the information flow graph that consists of source nodes and coding nodes. We define $T(p, N)$ to be the family of bipartite information flow graphs corresponding to the $ZK(p, N)$ network configurations. Receiver N observes only source nodes and always receives rate h, thus we can ignore it and consider the remaining $N - 1$ receivers. Let $\mathcal{I}' = \{1, 2, \ldots, N-1\}$ be an index set. We define two more index sets: $\mathcal{A}'$ as the set of all $(p-1)$-element subsets of $\mathcal{I}'$ and $\mathcal{C}$ as the set of all p-element subsets of $\mathcal{I}'$.

- We have $h = \binom{N-1}{p-1}$ source nodes indexed by the elements of $\mathcal{A}'$. Each source node is observed by the corresponding set of $p - 1$ receivers.
- We have $\binom{N-1}{p}$ coding nodes indexed by the elements of $\mathcal{B}'$. Each coding node is observed by the corresponding set of p receivers.
- A source node S is connected to a coding node T if the index S is a subset of the index of T.

It follows that

- Each source node has outgoing degree $N - p$ and each coding node has incoming degree p.
- Each receiver (except for receiver N) observes $x_1 = \binom{N-2}{p-2}$ source nodes and $x_2 = \binom{N-2}{p-1} = \frac{N-p}{p-1} x_1$ coding nodes.
- The configuration is symmetric with respect to receivers and sources.

Lemma 1: The family of information flow graphs $T(p, N)$ have the following properties.

1) Removing any edge of the node graph reduces the min-cut by one for exactly one receiver.

2) The resulting graph still has property 1).

Proof: Assume that receiver R_i observes the node T, and that node T is connected to the p source nodes $\{S_1,\dots S_p\}$. From construction receiver R_i observes $p-1$ of the $\{S_1,\dots S_p\}$ source nodes, say $S_1 \dots S_{p-1}$, and observes the remaining source, say source S_p, *only* through node T. As a result, removing the edge (T, S_p) will violate the min-cut property for receiver R_i. Moreover, the min-cut property for all other receivers will not be affected. ■

In Section II and in more detail in ([16], Def. 3) we defined a configuration to be minimal with the min-cut property if removing any edge will violate the min-cut property for at least one receiver. Note that this definition does not necessarily assume that the min-cut is the same for all receivers. Also note that, if we start from a minimal configuration, and remove an edge, then we get a configuration where the min-cut for one or more receivers is smaller, and the new configuration is not necessarily minimal, i.e., it may contain other edges which we can remove without further violating the min-cut condition. Lemma 1 tells us that the family $T(p,N)$ is minimal with the min-cut property, and moreover, removing any number of edges leads to a configuration that is again minimal.

Since the configuration $T(p,N)$ is minimal, to achieve throughput $h=\binom{N-1}{p-1}$ for all receivers, we need to employ $\binom{N-1}{p} = h\frac{N-p}{p}$ coding points. For example, for $p << N$, we need $\approx hN$ coding points. In a practical network, the coding points correspond to nodes in the network that have enhanced functionalities. Thus, we might have a restricted number of such nodes. In [20], the number of required coding points is termed encoding complexity, and it is shown that an upper bound is h^3N^2.

The following Lemma characterizes the trade-off between encoding complexity and achievable rate for the $T(p,N)$ configurations.

Lemma 2: For the family $T(p,N)$, if we allow k out of the $\binom{N-1}{p}$ coding points to perform linear combining, while the remaining coding points may only forward one of their incoming information flows, we can achieve average throughput T_k^{av} such that

$$\frac{T_k^{av}}{T_c} = \frac{1}{N-1}(p-1+\frac{N-p}{p}+k\frac{p-1}{h}). \tag{20}$$

Proof: If k out of the $\binom{N-1}{p}$ coding points are allowed to perform combining, we get that

$$T_k^{av} = \frac{1}{N-1}((N-1)\binom{N-2}{p-2}+\binom{N-1}{p}+k(p-1)). \tag{21}$$

This is because,

- Each of the $N-1$ receivers observes throughput $\binom{N-2}{p-2}$ directly at the source nodes.
- At each of the $\binom{N-1}{p}$ coding points, at least one receiver successfully receives rate 1.
- At the k coding points where coding is allowed, the remaining $p-1$ receivers also receive rate 1. This can be achieved by binary addition of the inputs at each coding point.

Using simple identities such as

$$\binom{N-2}{p-2} = \binom{N-1}{p-1}\frac{p-1}{N-1} = h\frac{p-1}{N-1} \text{ and } \binom{N-1}{p} = \binom{N-1}{p-1}\frac{N-p}{p} = h\frac{N-p}{p},$$

we get Eq. (20). ∎

Note that substituting $k = h\frac{N-p}{p}$ in Eq. (21), i.e., using network coding at all coding points, we get that $T_k^{av} = T_c = h$ as expected. At the other extreme, substituting $k = 0$, i.e., using routing only, we get an exact characterization of T_i^{av} as

$$T_k^{av} = T_i^{av} = \frac{h}{N-1}(p-1+\frac{N-p}{p}+k\frac{p-1}{h}).$$

This expression asymptotically coincides with the upper bound in Eq. (16).

Additionally, Lemma 2 shows that the throughput benefits increase *linearly* with the number of coding points k, at a slope of $\frac{p-1}{h(N-1)}$. Thus, a significant number of coding points are required to achieve a constant fraction of the network coding throughput.

VI. Conclusions

We have investigated benefits that network coding offers with respect to the average throughput achievable by routing, where the average throughput refers to the average of the rates that the individual receivers experience. It was shown that these benefits are related to the integrality gap of a standard LP formulation for the directed Steiner tree problem. Based on this connection, a class of directed graph configurations with N receivers for which network coding offers benefits proportional to $\sqrt{N}$ was identified. However, it was remarkable to see that for fairly large classes of networks, network coding at most doubles the average throughput. Several such classes were identified. A comparison between the average and other throughput measures used in network coding literature was addressed, often to point out the difference in coding benefits. It was shown that for certain classes of networks, the average throughput can be achieved uniformly by all receivers by employing vector routing and channel coding. Some issues concerning the network code alphabet size as a trade-off between routing and coding as well as between required for deterministic and randomized coding were addressed. It was shown, that for certain classes of networks, there are huge savings to be made in terms of alphabet size if one resorts to routing as opposed to coding with a small throughput loss, or to deterministic as opposed to random coding with no throughput loss.

Acknowledgment

The authors would like to thank Moses Charikar for useful discussions. This work was supported in part by the NSF under award No. CCR-0325673 and FNS under award No. 200021-103836/1.

References

[1] R. Ahlswede, N. Cai, S-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, pp. 1204–1216, July 2000.

[2] S-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, vol. 49, pp. 371–381, Feb. 2003.

[3] Z. Li and B. Li, "Network coding in undirected networks," *Proceeding of CISS 2004*, 2004.

[4] Y. Wu, P. A. Chou, and K. Jain, "A comparison of network coding and tree packing," *ISIT 2004*, 2004.

[5] P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial time algorithms for network information flow," *Proc. 15th ACM Symposium on Parallel Algorithms and Architectures*, 2003.

[6] A. Agarwal and M. Charikar, "On the advantage of network coding for improving network throughput," in *Proc. of 2004 IEEE Information Theory Workshop*, San Antonio, Texas, Oct. 2004.

[7] L. Zosin and S. Khuller, "On directed steiner trees," in *Proc. 13th Annual ACM/SIAM Symposium on Discrete Algorithms (SODA '02)*, pp. 59–63, 2002.

[8] E. Halperin, G. Kortsarz, R. Krauthgamer, A. Srinivasan, and N. Wang, "Integrality ratio for group steiner trees and directed steiner trees," in *Proc. of 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Jan. 2003.

[9] A. Rasala-Lehman and E. Lehman, "Complexity classifi cation of network information flow problems," *SODA*, pp. 142–150, 2004.

[10] R. Dougherty, C. Freiling, and K. Zeger, "Insuffi ciency of linear coding in network information flow," *Submitted to Trans. Inf. Theory*, 2004.

[11] M. Luby, M. Mitzenmacher, A. S. D. Spielman, and V. Stemann, "Practical loss-resilient codes," *ACM Symposium on Theory of Computing*, 1997.

[12] A. Shokrollahi, "Raptor codes," *Submitted to IEEE Trans. Information Theory*, 2004.

[13] J. Cannons, R. Dougherty, C. Freiling, and K. Zeger, "Network routing capacity," *Submitted to IEEE/ACM Trans. on Networking.*, 2004.

[14] S. Jaggi, P. Chou, and K. Jain, "Low complexity algebraic multicast network codes," *Proc. IEEE International Symposium Information Theory*, p. 368, 2003.

[15] T. Ho, M. Mèdard, J. Shi, M. Effros, and D. R. Karger, "On randomized network coding," *Proc. 41st Annual Allerton Conference*, Monticello, IL, Oct. 2003.

[16] C. Fragouli and E. Soljanin, "Information flow decomposition for network coding," *IEEE Trans. Inform. Theory*, July 2004.

[17] A. Schrijver, *Theory of Linear and Integer Programming*. John Wiley and Sons, 1986.

[18] M. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, eds., *Network Models*. North Holland, 1994.

[19] V. F. Kolchin, B. A. Sevastianov, and V. P. Christiakov, *Random Allocations*. Wiley, John & Sons, Inc., 1978.

[20] M. Langberg, A. Sprintson, and J. Bruck, "The encoding complexity of network coding," *Proc. IEEE International Symposium Information Theory*, Sept 2005.